

APPENDIX XV

OPEN_VLAN_PORT(Vlan)

Return any unused Vlan port number

ENABLE_VLAN_PORT(Vlan, VlanPort, ProtocolSpecifier)

(*should check whether Routing type of Vlan server matches ProtocolSpecifier *)

(* and not allow the port to be opened if there no match. Not required initially *)

(* when we will only have one Routing type. *)

(* If Vlan is unknown we add a mapping for every server link *)

For each L in FindLinks(Vlan) do

(* look through links of Vlan server *)

LP:= FindLinkPortofProtocol(ProtocolSpecifier, L)

If LP = Nil then

(* no link port for protocol specifier *)

LP := OpenLinkPort(L)

(* create a link port on link L *)

EnableLinkPort(LP, ProtocolSpecifier)(*enable receipt of packets matching specifier
on LP*)

AddPortMapping(Vlan, VlanPort, L, LP, ProtocolSpecifier) (*)

TRANSMIT(B,DA, Vlan,VlanPort)

(*transmit buffer B to Destination DA on Vlan*)

Let VR be the VlanRecord corresponding to Vlan

If VR.State = ON then

(* never transmit on an incorrectly set up Vlan *)

Let SR be the server record corresponding to Vlan.ServerName

L = Split(SR.LiveLinks)

(* choose one link from set of live links to Vlan Server *)

(* next pick source address equal to unique ID of assigned link for this Vlan*)

SA:= Source 48-bit Address of VR.AssignedLink

(*source address of packet*)

Create a Data Link packet P from buffer B with Destination Address DA and Source

Address SA

(* next add a VlanId only if multicast or going to a VML Client*)

If (DA = Multicast) or (DA is in LinkArray[L].ClientAddresses) then P =

AddVlanID(P,Vlan)

LP:= FindLinkPortofVlan(Vlan,VlanPort,L); (* find link port for Vlan and VlanPort on link L *)

Enqueue packet P to link port LP (* transmit to link port on chosen link *)

RECEIVE (P, L, LP) (* packet P received on link L and data link port LP *)

If P is a multicast packet then V = VlanfromPacket(P) (*if multicast find Vlan from VlanId in packet*)

Else (* if not multicast, look up source address of packet in mapping table *)

V = SourceLookup(P.SA)

If V = VMLRouterId then then V = VlanfromPacket(P);

(* Note that V may contain the VlanId of the Unknown Vlan *)

VP = FindVlanPort(V,L,LP) (* find Vlan port corresponding to Vlan V *)

Enqueue packet P to Virtual LAN Port VP